



# The multi-item capacitated lot-sizing problem with setup times and shortage costs

Nabil Absi, Safia Kedad-Sidhoum

## ► To cite this version:

Nabil Absi, Safia Kedad-Sidhoum. The multi-item capacitated lot-sizing problem with setup times and shortage costs. *European Journal of Operational Research*, 2008, 185 (3), pp.1351-1374. 10.1016/j.ejor.2006.01.053 . hal-00506261

**HAL Id: hal-00506261**

**<https://hal.science/hal-00506261>**

Submitted on 27 Jul 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The multi-item capacitated lot-sizing problem with setup times and shortage costs

Nabil Absi<sup>1,2\*</sup>, Safia Kedad-Sidhoum<sup>1†</sup>

<sup>1</sup>Laboratoire LIP6, 4 place Jussieu, 75 252 Paris Cedex 05, France

<sup>2</sup>Dynasys S.A., 10 Avenue Pierre Mendes France, 67 300 Schiltigheim, France

## Abstract

We address a multi-item capacitated lot-sizing problem with setup times and shortage costs that arises in real-world production planning problems. Demand cannot be backlogged, but can be totally or partially lost. The problem is NP-hard. A mixed integer mathematical formulation is presented. Our approach in this paper is to propose some classes of valid inequalities based on a generalization of Miller *et al.* [26] and Marchand and Wolsey [24] results. We also describe fast combinatorial separation algorithms for these new inequalities. We use them in a branch-and-cut framework to solve the problem. Some experimental results showing the effectiveness of the approach are reported.

**Keywords:** multi-item, capacitated lot-sizing, setup times, shortage costs, production planning, mixed integer programming, branch-and-cut.

## Introduction

The Multi-item Capacitated Lot-sizing Problem with Setup times and Shortage costs called MCLSSP is a production planning problem in which there is a time-varying demand for a set of  $N$  items denoted  $\mathcal{I} = \{1, 2, \dots, N\}$  over  $T$  periods. The production should satisfy a restricted capacity and must take into account a set of additional constraints. Indeed, launching the production of an item  $i$  at a given period  $t$  for a demand requirement  $d_{it}$  involves a variable capacity  $v_{it}$  and a fixed consumption of resource  $f_{it}$  usually called setup time in lot-sizing literature. The total available capacity at period  $t$  is  $c_t$ . The production should also satisfy lot-sizing constraints. For each period  $t$ , an inventory cost  $\gamma_{it}$  is attached to each item  $i$  as well as a variable unit production cost  $\alpha_{it}$  and a setup cost  $\beta_{it}$ . The problem has the distinctive feature of allowing requirement shortages because we deal with problems with tight capacities. Indeed, when we are in lack of capacity to produce the total demand, we try to spread the capacity among the items by minimizing the total amount of demand shortages. Thus, we introduce in the model a unit cost parameter  $\varphi_{it}$  for item  $i$  at period  $t$  for the requirement not met regarding the demand. These costs should be viewed as penalty costs and their values are very high in comparison with other cost components.

To try to meet the demand for an item  $i$  at period  $t$ , we could anticipate the production over some periods of time. Therefore,  $\sigma_{it}$  denote the last period at which an item  $i$  produced at period  $t$  can be consumed.

The problem MCLSSP is to find a production planning that minimizes the demand shortage, the setup, the inventory and the production costs. Originally, the motivation for designing a branch-and-cut algorithm to solve the MCLSSP was to try to deal with real-world instances where the capacities were tight and where the most important objective was to try to meet the maximum

---

\*This work has been partially financed by DYNASYS S.A., under research contract no. 588/2002.

†Corresponding author. E-mail: Safia.Kedad-Sidhoum@lip6.fr

amount of client's needs. In these industrial applications, postponing the demand is frequently prohibited. Our results are integrated in an APS<sup>1</sup> software.

Florian *et al.* [15] and Bitran and Yanasse [7] have shown that the single-item capacitated lot-sizing problem is NP-hard, even for many special cases. Chen and Thizy [10] have proved that multi-item capacitated lot-sizing problem (MCLSP) with setup times is strongly NP-hard.

Since the seminal papers by Wagner and Within [38] and Manne [23] in the late 1950s, a lot of research has been done on lot-sizing problems. The single-item problem has been given special interest for its relative simplicity and for its importance as a sub-problem of some more complex lot-sizing problems. For a complete review, the reader can refer to [9].

Although production planning models involving multiple items, restrictive capacities and significant setup times occur frequently in industrial situations and have often been studied in the literature, obtaining optimal and sometimes even feasible solutions remains challenging. Trigeiro *et al.* [36] were among the firsts to try to solve such models. They proposed a lagrangean relaxation based heuristic to solve the single-machine, multi-item, capacitated lot-sizing problem with setup times to obtain near-optimal solutions. Since the lagrangean solutions are usually infeasible, they used a smoothing heuristic in order to obtain feasible production plans. However, we can notice that for all the instances with tight capacities, they were not able to find feasible solutions.

Belvaux and Wolsey [6], Leung *et al.* [19] and Pochet and Wolsey [32] proposed exact methods to solve multi-item capacitated lot-sizing problems by strengthening the LP formulations with valid inequalities and then using a mixed integer programming (MIP) solver. Barany *et al.* [4] have defined some inequalities for the uncapacitated lot-sizing problem. Miller *et al.* [26] have studied the polyhedral structure of some capacitated production planning problems with setup times. We can also mention the work of Marchand and Wolsey [24] for the 0-1 knapsack problem which appears as a relaxation of a number of structured MIP problems such as the MCLSP problem.

There are few references dealing with lot-sizing problems with shortage costs. Recently, Sandbothe and Thompson [35] addressed a single-item uncapacitated lot-sizing problem with shortage costs. The authors proposed an  $O(T^3)$  forward dynamic programming algorithm to solve the problem. Aksent *et al.* [3] proposed a dynamic programming method to solve the same problem in  $O(T^2)$ . Loparic *et al.* [21] proposed valid inequalities for the single-item uncapacitated lot-sizing problem with sales instead of fixed demands and lower bounds on stock variables. To the best of our knowledge, this is the first paper that deals with setup times constraints and shortage costs for the multi-item capacitated lot-sizing problem. Nevertheless, we can cite the following results for solving problems where demand cannot be met at every period. Dixon *et al.* [13] deal with lack of capacity by considering overtime. The capacity constraint is expanded by making extra capacity available at a certain cost. The multi-item capacitated lot-sizing problem with setup times and overtime decisions is investigated by Diaby *et al.* [12], Özdamar and Birbil [28] and Özdamar and Bozyel [29]. Another class of methods allows backlog. Here demand must be satisfied, but the items can be produced later at an extra cost. We can cite the work of Pochet and Wolsey [31] and Zangwill [41]. In all these cases, the demand must be satisfied and the amount of lost sales for each item at each period is not given. The only information that we have is the amount of missing capacity at each period to satisfy the amount of original and backlogged demands.

The main contributions of this paper are twofold. First, we show that the results obtained from considering relaxations based on single-period sub-model can be used to derive new valid inequalities for the MCLSP problem. These results are derived from Miller *et al.* [26] previous work on the polyhedral structure of the single-period relaxation of the multi-item capacitated lot-sizing problem. Second, we use these inequalities within a branch-and-cut framework to find near optimal solutions.

An outline of the remainder of the paper follows. Sections 1 and 2 describe MIP formulations of the MCLSP problem and its single-period relaxation. In Sections 3 and 5 we state results concerning the generalization of the  $(l, S)$ , cover and reverse cover valid inequalities. In Section 6, we show that these inequalities can be strengthened using a lifting procedure. Separation heuristics are presented in Sections 4, 7 and 8. Finally, computational results are given in Section 9 to show the effectiveness of using these inequalities in a branch-and-cut algorithm.

---

<sup>1</sup>Advance Planning and Scheduling.

# 1 Formulation of the MCLSSP problem

In this section we present a MIP formulation of the MCLSSP problem, which is an extension of the classical formulation of the MCLSP problem previously studied by Miller [25] and Trigeiro *et al.* [36]. This model is usually called aggregated model, see [9]. Other formulations are studied in the literature. We can mention the facility location-based formulation introduced by Krarup and Bilde [18] and the shortest path formulation proposed by Evans [14].

In the sequel of the paper, we consider that  $i = 1, \dots, N$  and  $t = 1, \dots, T$ . We set  $x_{it}$  as the quantity of item  $i$  produced at period  $t$ . To deal with the fixed setup times and costs, we need also to define  $y_{it}$  as a binary variable equal to 1 if item  $i$  is produced at period  $t$  (i.e. if  $x_{it} > 0$ ). The variable  $s_{it}$  is the inventory value for item  $i$  at the end of period  $t$ . The demand shortage for item  $i$  at period  $t$  is modeled by a non-negative variable  $r_{it}$  added to the production variables  $x_{it}$  with a very high unit penalty cost in the objective function, because the main goal is to satisfy the customer and thus to have the minimum amount of the requirements not met. We can notice that  $r_{it} = -(s_{i,t-1} + x_{it}) + d_{it}$  if  $r_{it} > 0$  and 0 otherwise.

$$\min \sum_{i=1}^N \sum_{t=1}^T \alpha_{it} x_{it} + \beta_{it} y_{it} + \gamma_{it} s_{it} + \varphi_{it} r_{it} \quad (1)$$

subject to:

$$x_{it} + r_{it} - s_{it} + s_{i,t-1} = d_{it}, \quad i = 1, \dots, N, t = 1, \dots, T. \quad (2)$$

$$\sum_{i=1}^N v_{it} x_{it} + \sum_{i=1}^N f_{it} y_{it} \leq c_t, \quad t = 1, \dots, T. \quad (3)$$

$$x_{it} \leq \min \left\{ \frac{c_t - f_{it}}{v_{it}}, \sum_{t'=t}^{\sigma_{it}} d_{it'} \right\} y_{it}, \quad i = 1, \dots, N, t = 1, \dots, T. \quad (4)$$

$$r_{it} \leq d_{it}, \quad i = 1, \dots, N, t = 1, \dots, T \quad (5)$$

$$x_{it}, s_{it}, r_{it} \geq 0, \quad i = 1, \dots, N, t = 1, \dots, T \quad (6)$$

$$y_{it} \in \{0, 1\}, \quad i = 1, \dots, N, t = 1, \dots, T \quad (7)$$

The objective function (1) minimizes the total cost induced by the production plan (unit production costs, inventory costs, shortage costs and setup costs). Constraints (2) are the flow conservation of the inventory through the planning horizon. Constraints (3) are the capacity constraints, the overall consumption must remain lower than the available capacity. If we produce an item then the production must not exceed a maximum production level, this condition is ensured by constraints (4). Indeed, the maximum production is the minimum between the maximum quantity of the item that we can produce and the total requirement on section  $[t, \dots, \sigma_{it}]$  of the horizon. We recall that  $\sigma_{it}$  denote the last period at which an item  $i$  produced at period  $t$  can be consumed. Constraints (5) define upper bounds on the requirement not met for item  $i$  on period  $t$ . Constraints (6) and (7) characterize the variable's domain:  $x_{it}$ ,  $s_{it}$  and  $r_{it}$  are non-negative for  $i = 1, \dots, N$  and  $t = 1, \dots, T$  and  $y_{it}$  is a binary variable for  $i = 1, \dots, N$  and  $t = 1, \dots, T$ .

In the sequel of the paper, we refer to valid inequalities for the set defined by (2) – (7) as valid for MCLSSP.

## 2 Single-period relaxation of the MCLSSP problem

Based on the formulation of the MCLSSP problem described in section 1, we define a simplified sub-model obtained by considering a single time period relaxation. This is particularly useful to derive valid inequalities for the MCLSSP problem. The goal of this relaxation is not to solve each period separately by considering only the demand of the current period but is to provide strong valid inequalities for the single-period problem that are also valid for the initial problem taking into account aggregated demands. This is done by allowing anticipations on production.

This model is called the single-period relaxation of the MCLSSP with preceding inventory [25]. Our approach is similar to the one used by Constantino [11] and Miller [25] to derive a set of valid inequalities for the MCLSSP problem based on a single-period relaxation.

In this relaxation, the production over a given period could satisfy the requirement of a section of consecutive periods. Consequently, for each period  $t = 1, \dots, T$  and each item  $i = 1, \dots, N$  we use the parameter  $\sigma_{it}$  previously defined with  $\sigma_{it} = 1, \dots, T$ . This will enable us to create a mathematical model for each period  $t = 1, \dots, T$  which captures the interaction between the tight capacity in one hand and the requirements, the productions and the setups on the other hand from period  $t$  to  $\sigma_{it}$ , for each item  $i = 1, \dots, N$ .

Here our goal is to derive valid inequalities for MCLSSP by considering simplified models obtained from a single time period relaxation with preceding inventory.

Let us denote:  $\delta_{a,b}^i = \sum_{t=a}^b d_{it}$ . One simple family of valid inequalities is given by

**Proposition 1.** *The inequalities*

$$x_{it} + \sum_{t'=t}^{\sigma_{it}} r_{it'} + \left( s_{i,t-1} + \sum_{t'=t+1}^{\sigma_{it}} \delta_{t',\sigma_{it}}^i y_{it'} \right) \geq \delta_{t,\sigma_{it}}^i, \quad i = 1, \dots, N, t = 1, \dots, T. \quad (8)$$

are valid for MCLSSP.

*Proof.* Summing the constraints (2) over the section of horizon  $[t, \dots, \sigma_{it}]$  gives:

$$\sum_{t'=t}^{\sigma_{it}} (x_{it'} + r_{it'}) - s_{i,\sigma_{it}} + s_{i,t-1} = \sum_{t'=t}^{\sigma_{it}} d_{it'}, \quad i = 1, \dots, N, t = 1, \dots, T. \quad (9)$$

The variable  $x_{it}$  can be redefined by considering the period where the production is really consumed. This reformulation is called the facility location-based formulation introduced initially by Krarup and Bilde [18]. Therefore, we denote  $w_{itt'}$  with  $t' \in [t, \sigma_{it}]$  the quantity of the item  $i$  produced at period  $t$  ( $t \neq 0$ ) and consumed at period  $t'$ . The variables  $w_{i0t}$  then represent the opening inventory of item  $i$  at the beginning of the horizon which will be consumed at period  $t$ . We will have:

$$x_{it} = \sum_{t'=t}^T w_{itt'}, \quad i = 1, \dots, N, t = 1, \dots, T. \quad (10)$$

and

$$s_{it} = \sum_{t'=0}^t \sum_{t''=t+1}^T w_{it't''}, \quad i = 1, \dots, N, t = 1, \dots, T. \quad (11)$$

By replacing (10) and (11) in (9), we get for each  $i = 1, \dots, N$  and  $t = 1, \dots, T$ :

$$s_{i,t-1} + x_{it} + \sum_{t'=t+1}^{\sigma_{it}} \sum_{t''=t'}^T w_{it't''} + \sum_{t'=t}^{\sigma_{it}} r_{it'} - \sum_{t'=0}^{\sigma_{it}} \sum_{t''=\sigma_{it}+1}^T w_{it't''} = \sum_{t'=t}^{\sigma_{it}} d_{it'} \quad (12)$$

Moreover:

$$\sum_{t'=t+1}^{\sigma_{it}} \sum_{t''=t'}^T w_{it't''} = \sum_{t'=t+1}^{\sigma_{it}} \sum_{t''=t'}^{\sigma_{it}} w_{it't''} + \sum_{t'=t+1}^{\sigma_{it}} \sum_{t''=\sigma_{it}+1}^T w_{it't''} \quad (13)$$

and:

$$\sum_{t'=0}^{\sigma_{it}} \sum_{t''=\sigma_{it}+1}^T w_{it't''} = \sum_{t'=0}^t \sum_{t''=\sigma_{it}+1}^T w_{it't''} + \sum_{t'=t+1}^{\sigma_{it}} \sum_{t''=\sigma_{it}+1}^T w_{it't''} \quad (14)$$

By replacing (13) and (14) in (12), we get for each  $i = 1, \dots, N$  and  $t = 1, \dots, T$ :

$$s_{i,t-1} + x_{it} + \sum_{t'=t+1}^{\sigma_{it}} \sum_{t''=t'}^{\sigma_{it}} w_{it't''} - \sum_{t'=0}^t \sum_{t''=\sigma_{it}+1}^T w_{it't''} + \sum_{t'=t}^{\sigma_{it}} r_{it'} = \sum_{t'=t}^{\sigma_{it}} d_{it'} \quad (15)$$

By definition of variables  $w_{it't''}$ , we know that:

1.  $w_{it't''} \leq d_{it''} y_{it'}$
2.  $w_{i0t} \leq d_{it}$
3.  $\sum_{t'=0}^t \sum_{t''=\sigma_{it}+1}^T w_{it't''} = 0$

Consequently, from (15), we get for each  $i = 1, \dots, N$  and  $t = 1, \dots, T$ :

$$s_{i,t-1} + x_{it} + \sum_{t'=t+1}^{\sigma_{it}} \sum_{t''=t'}^{\sigma_{it}} d_{it''} y_{it'} + \sum_{t'=t}^{\sigma_{it}} r_{it'} \geq \sum_{t'=t}^{\sigma_{it}} d_{it'}$$

Furthermore,

$$\sum_{t''=t'}^{\sigma_{it}} d_{it''} = \delta_{t', \sigma_{it}}^i$$

Finally,

$$x_{it} + \sum_{t'=t}^{\sigma_{it}} r_{it'} + s_{i,t-1} + \sum_{t'=t+1}^{\sigma_{it}} \delta_{t', \sigma_{it}}^i y_{it'} \geq \delta_{t, \sigma_{it}}^i$$

□

In the sequel of the paper, we denote by SPMCLSSP the Single-Period relaxation of the problem MCLSSP where (2) is replaced by (8). As previously mentioned, we refer to valid inequalities for the set defined by (3) – (8) as valid for SPMCLSSP.

The expression  $s_{i,t-1} + \sum_{t'=t+1}^{\sigma_{it}} \delta_{t', \sigma_{it}}^i y_{it'}$  can be considered as being the ending inventory of item  $i$  at period  $t - 1$  and denoted  $\tilde{s}_{i,t-1}$ . Thus, we have  $\tilde{s}_{i,t-1} = s_{i,t-1} + \sum_{t'=t+1}^{\sigma_{it}} \delta_{t', \sigma_{it}}^i y_{it'}$ . Similarly, we note  $\sum_{t'=t}^{\sigma_{it}} r_{it'}$  by  $\tilde{r}_{it}$  and  $\delta_{t, \sigma_{it}}^i$  by  $\tilde{d}_{it}$ . The inequalities (8) are equivalent to:

$$x_{it} + \tilde{r}_{it} + \tilde{s}_{it} \geq \tilde{d}_{it}, \quad i = 1, \dots, N, t = 1, \dots, T. \quad (16)$$

Since we work on a single-period in SPMCLSSP and given that each period will be considered separately, it may be more convenient to remove the temporal index in the previous expression to facilitate the reading of the remaining mathematical formulations.

The inequalities (8) are written:

$$x_i + \tilde{r}_i + \tilde{s}_i \geq \tilde{d}_i, \quad i = 1, \dots, N. \quad (17)$$

### 3 Valid $(l, S)$ inequalities for the problem SPMCLSSP

To introduce the  $(l, S)$  inequalities for SPMCLSSP, let us define the following problem denoted MCLSP by:

$$\min \sum_{i=1}^N \sum_{t=1}^T \alpha_{it} x_{it} + \beta_{it} y_{it} + \gamma_{it} s_{it} \quad (18)$$

subject to:

$$x_{it} - s_{it} + s_{i,t-1} = d_{it}, \quad i = 1, \dots, N, t = 1, \dots, T \quad (19)$$

$$\sum_{i=1}^N x_{it} + \sum_{i=1}^N f_{it} y_{it} \leq c_t, \quad t = 1, \dots, T. \quad (20)$$

$$x_{it}, s_{it} \geq 0, \quad i = 1, \dots, N, t = 1, \dots, T \quad (21)$$

$$y_{it} \in \{0, 1\}, \quad i = 1, \dots, N, t = 1, \dots, T \quad (22)$$

The problem MCLSP is a simplified version of MCLSSP with  $v_{it}$  equal to 1 and no demand shortage allowed, so that the variables  $r_{it}$  are set to zero.

We denote:

- SPMCLSP the single-period relaxation of MCLSP with (19) replaced by:  $x_i + s_i \geq d_i$ . We recall that the temporal index is removed.
- ULSP the uncapacitated version of the single-item relaxation of MCLSP. In this problem, the capacity constraints linking the items are removed. Thus, each item is considered separately and the item index is useless.

Barany *et al.* [4, 5] proved that a complete polyhedral description of the convex hull of the ULSP is given by some inequalities from the basic LP relaxation of the standard MIP formulation together with the  $(l, S)$  inequalities. The  $(l, S)$  inequalities are expressed as :

$$\sum_{t \in S} x_t + \sum_{t \in \bar{S}} d_{it} y_t \leq d_{il}$$

Where  $l \in \{0, 1, \dots, T\}$ ,  $S \subset \{0, 1, \dots, l\}$ ,  $\bar{S} = \{0, 1, \dots, l\} \setminus S$  and  $d_{it'} = \sum_{k=t}^{t'} d_k$ . The authors reported good computational results for multi-item capacitated lot-sizing problems using the  $(l, S)$  inequalities within a branch-and-cut scheme.

**Proposition 2.**  *$(l, S)$  inequalities for the SPMCLSP, (Miller *et al.* [26])*

*If  $c \geq f_i + d_i$  then the inequalities*

$$s_i + d_i y_i \geq d_i, \quad i = 1, \dots, N \quad (23)$$

*are facet-inducing for the SPMCLSP.*

Pochet and Wolsey [33] introduced the  $(k, l, S, I)$  inequalities for the single-item lot-sizing problem with constant capacity and they showed that these are nontrivial facets of its convex hull. The  $(k, l, S, I)$  inequalities can be expressed in the following general form:

$$s_{t-1} + \sum_{t \in U} x_t + \sum_{t \in V} B_t y_t \geq B_0$$

where for any  $k$  and  $l$  such that  $1 \leq k \leq l \leq T$ ,  $(U, V)$  is any partition of  $[k, \dots, l]$ ,  $B_0 \in \mathbb{R}_+$  and  $B_t \in \mathbb{R}_+$  ( $t \in V$ ).  $B_t$  are defined with respect to the demand and the capacity. For more details, the reader can refer to [33].

Without loss of generality, we can modify the inequalities (8) in order to have a structure similar to the  $(k, l, S, I)$  inequalities, by replacing  $\delta_{t', \sigma_{it}}^i y_{it'}$  by  $x_{it'}$  for values of  $t'$  in any subset of  $[t+1, \dots, \sigma_{it}]$ .

**Proposition 3.** *Given a partition  $(U, V)$  of the interval  $[t+1, \dots, \sigma_{it}]$ , the inequalities*

$$x_{it} + \sum_{t'=t}^{\sigma_{it}} r_{it'} + \left( s_{i,t-1} + \sum_U \delta_{t', \sigma_{it}}^i y_{it'} + \sum_V x_{it'} \right) \geq \delta_{t, \sigma_{it}}^i, \quad i = 1, \dots, N, t = 1, \dots, T \quad (24)$$

*are valid for MCLSSP.*

*Proof.* The proof is similar to the proof of proposition 1. □

The inequalities (24) are called the  $(l, S)$  inequalities for the problem SPMCLSSP.

## 4 Separation heuristic for $(l, S)$ inequalities

In this section, we present a fast combinatorial separation heuristic to create  $(l, S)$  inequalities for the MCLSSP problem. According to the proposition 3, the  $(l, S)$  inequalities (24) are valid for MCLSSP. We recall the expression of these inequalities:

$$x_{it} + \sum_{t'=t}^{\sigma_{it}} r_{it'} + \left( s_{i,t-1} + \sum_U \delta_{t', \sigma_{it}}^i y_{it'} + \sum_V x_{it'} \right) \geq \delta_{t, \sigma_{it}}^i, \quad i = 1, \dots, N, t = 1, \dots, T$$

with  $(U, V)$  a partition of  $[t+1, \dots, \sigma_{it}]$ .

The idea of the separation heuristic is to create a set  $U \subset [t, \dots, \sigma_{it}]$  for each item  $i$  and for each period  $t$  for generating an  $(l, S)$  inequality for the MCLSSP problem. We add  $t'$  to  $U$  if  $\delta_{t', \sigma_{it}}^i y_{it'} < x_{it'}$  or to the set  $V$  otherwise. We illustrate this principle in the following algorithm:

**Algorithm 1** Separation heuristic for  $(l, S)$  inequalities

```

1:  $t \leftarrow 1, i \leftarrow 1$ 
2: while  $(i \leq N)$  do
3:   while  $(t \leq T)$  do
4:      $t' \leftarrow t + 1$ 
5:     while  $(t' \leq \sigma_{it})$  do
6:       if  $\delta_{t', \sigma_{it}}^i y_{it'}^* < x_{it'}^*$  then
7:          $U \leftarrow U \cup \{t'\}$ 
8:       else
9:          $V \leftarrow V \cup \{t'\}$ 
10:      end if
11:       $t' \leftarrow t' + 1$ 
12:    end while
13:    if (The inequality (24) based on  $S, U$  and  $T'$  is violated) then
14:      Add the inequality (24) at the current node.
15:    end if
16:     $t \leftarrow t + 1$ 
17:  end while
18:   $i \leftarrow i + 1$ 
19: end while
```

We can notice that the separation heuristic for the  $(l, S)$  inequalities is in  $O(NT^2)$ .



## 5 Cover and reverse cover inequalities for the SPM-CLSSP

In this section, we generalize some results on the cover and reverse cover inequalities defined by Miller *et al.* [26].

**Definition 1.** (*Cover*)

A subset of items  $S$  of  $\mathcal{I}$  is known as "cover" of the problem SPMCLSSP if:

$$\lambda_S = \sum_{i \in S} (f_i + v_i \tilde{d}_i) - c \geq 0 \quad (25)$$

For the cover  $S$ ,  $\lambda_S$  expresses the lack of capacity when all the items of  $S$  are produced. Indeed, if  $\lambda_S > 0$  then the total requirements of all the items of  $S$  are strictly higher than the available capacity.

**Proposition 4.** (*Cover inequalities*)

The inequality

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} (1 - y_i) \quad (26)$$

is valid for SPMCLSSP.

*Proof.* The proof is similar to the one presented in Miller *et al.* [26] by adding the demand shortage variables  $\tilde{r}_i$  as well as the variable resource consumption  $v_i$ .

The inequalities (17) can be written:

$$\tilde{s}_i + \tilde{r}_i \geq \tilde{d}_i - x_i, \quad i = 1, \dots, N.$$

Then:

$$\sum_{i \in S} v_i \tilde{s}_i + \sum_{i \in S} v_i \tilde{r}_i \geq \sum_{i \in S} v_i \tilde{d}_i - \sum_{i \in S} v_i x_i$$

If all the items of  $S$  are produced,  $y_i = 1 \forall i \in S$ , from (3) we get:

$$\sum_{i \in S} v_i x_i \leq c - \sum_{i \in S} f_i$$

Then:

$$\sum_{i \in S} v_i \tilde{s}_i + \sum_{i \in S} v_i \tilde{r}_i \geq \sum_{i \in S} v_i \tilde{d}_i - \left( c - \sum_{i \in S} f_i \right) = \sum_{i \in S} (v_i \tilde{d}_i + f_i) - c$$

By replacing  $\sum_{i \in S} (v_i \tilde{d}_i + f_i) - c$  by  $\lambda_S$  we get:

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S \quad (27)$$

We define a set  $S^0 = \{i \in S : y_i = 0\}$  that represents the items in  $S$  that are not produced.

If  $|S^0| = 1$ , we have exactly one item  $i' \in S$  such that  $y_{i'} = 0$ .

From (27) we can write:

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S - f_{i'} \quad (28)$$

We know that:

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq v_{i'} (\tilde{s}_{i'} + \tilde{r}_{i'}) \geq v_{i'} \tilde{d}_{i'} \quad (29)$$

Thus, from (28) and (29) we can conclude that:

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S + \max \left\{ -f_{i'}, v_{i'} \tilde{d}_{i'} - \lambda_S \right\} \quad (30)$$

Let us consider now the case where  $|S^0| > 1$ . The inequality (30) can easily be generalized by considering the items in  $S^0$  one by one. Hence, we get:

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S + \sum_{i \in S^0} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} \quad (31)$$

The inequality (31) can be generalized for the set  $S$  by introducing the term  $(1 - y_i)$  to take into account the production of the item. Hence, we have:

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} (1 - y_i)$$

□

The previous inequalities can be strengthened by a lifting procedure described in what follows.

**Proposition 5.** (*A second form of cover inequalities*)

Given a cover  $S$  of SPMCLSSP, and an order of items  $i \in S$  such that  $f_{[1]} + v_{[1]} \tilde{d}_{[1]} \geq \dots \geq f_{[|S|]} + v_{[|S|]} \tilde{d}_{[|S|]}$ . Let  $T = \mathcal{I} \setminus S$ , and  $(T', T'')$  be any partition of  $T$ . We define  $\mu_1 = f_{[1]} + v_{[1]} \tilde{d}_{[1]} - \lambda_S$ . If  $|S| \geq 2$  and  $f_{[2]} + v_{[2]} \tilde{d}_{[2]} \geq \lambda_S$ , the inequality

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} (1 - y_i) + \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} \sum_{i \in T'} (v_i x_i - (\mu_1 - f_i) y_i) \quad (32)$$

is valid for SPMCLSSP.

*Proof.* Let  $(x^*, y^*, \tilde{s}^*, \tilde{r}^*)$  any point of the convex hull of SPMCLSSP. Let  $S^0 = \{i \in S : y_i^* = 0\}$ ,  $S^1 = \{i \in S : y_i^* = 1\}$  and  $\bar{T}' = \{i \in T' : y_i^* = 1\}$ . We consider three cases:

If  $|\bar{T}'| = 0$ , then we have from proposition (4) that the inequality (32) is valid.

If  $|\bar{T}'| = 1$ , then we assume that  $\bar{T}' = \{i'\}$ ; to show that the point  $(x^*, y^*, \tilde{s}^*, \tilde{r}^*)$  satisfies the inequality (32), it is sufficient to show that:

$$\begin{aligned} \sum_{i \in S} v_i (\tilde{s}_i^* + \tilde{r}_i^*) + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} y_i^* &\geq \lambda_S + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} + \\ &\quad \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} (v_{i'} x_{i'}^* - (\mu_1 - f_{i'})) \end{aligned} \quad (33)$$

We know that:

$$\sum_{i \in S} v_i (\tilde{s}_i^* + \tilde{r}_i^*) + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} y_i^* \geq \min_{x_{i'} = x_{i'}^*, y_{i'} = 1} \left\{ \sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} y_i \right\}$$

Let us consider the following problem:

$$\min_{x_{i'} = x_{i'}^*, y_{i'} = 1} \left\{ \sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} y_i \right\} \quad (34)$$

We will prove that the optimal solution of this problem has a value higher or equal to the right member of the inequality (33).

Let  $r_S = \left| \left\{ i \in S : f_i + v_i \tilde{d}_i > \lambda_S \right\} \right|$  and  $A_a^b = \sum_a^b (f_{[i]} + v_{[i]} \tilde{d}_{[i]})$ .

If we define:

$$\varphi_S(u) = \begin{cases} u, & \text{if } 0 \leq u \leq A_{r_S+1}^{[S]} \\ A_{r_S+1}^{[S]} + (r_S - j) \lambda_S, & \text{if } A_{j+1}^{[S]} \leq u \leq A_j^{[S]} - \lambda_S, j = 1, \dots, r_S \\ A_{r_S+1}^{[S]} + (r_S - j) \lambda_S + \left( u - (A_j^{[S]} - \lambda_S) \right), & \text{if } A_j^{[S]} - \lambda_S \leq u \leq A_j^{[S]}, j = 2, \dots, r_S \end{cases} \quad (35)$$

then the optimal solution of the minimization problem (34) is given by:

$$\min_{x_{i'} = x_{i'}^*, y_{i'} = 1} \left\{ \sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} y_i \right\} = \sum_{i \in S} v_i \tilde{d}_i + \varphi_S(c - (f_{i'} + v_{i'} x_{i'}^*)) \quad (36)$$

The proof is an obvious generalization of the result presented in [26]. We refer the reader to the paper of Miller *et al.* [26] for more details. A simple representation of  $\varphi_S$  is given in Fig. 1.

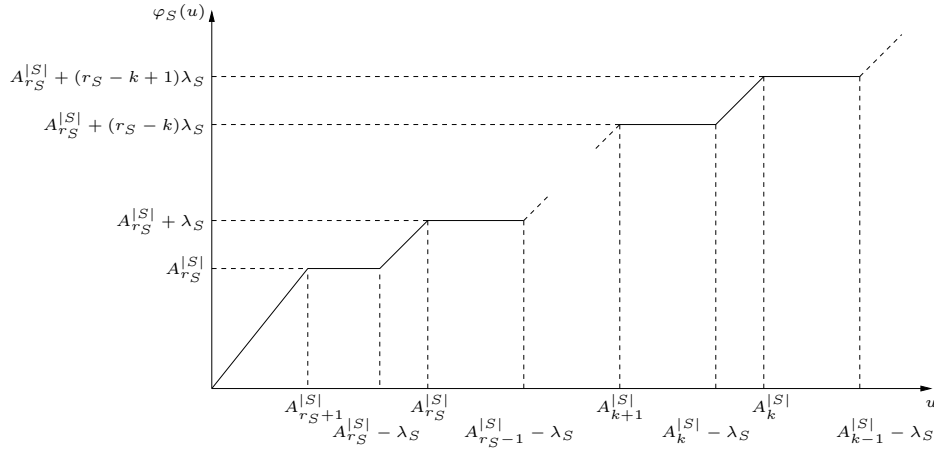


Figure 1: Function  $\varphi_S$ .

Now, we need to prove that:

$$\sum_{i \in S} v_i \tilde{d}_i - \varphi_S(c - (f_{i'} + v_{i'} x_{i'}^*)) \geq \lambda_S + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} + \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} (v_{i'} x_{i'}^* - (\mu_1 - f_{i'})) \quad (37)$$

To do that, we use the following property, which is also a generalization of a result presented in Miller *et al.* [26]):

$$\max_{0 \leq x_i \leq \frac{c-f_i}{v_i}} \left\{ \varphi_S(c - (f_i + v_i x_i)) + \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} (v_i x_i - (\mu_1 - f_i)) \right\} = \sum_{i \in S \setminus [1]} \min \left\{ f_i + v_i \tilde{d}_i, \lambda_S \right\} \quad (38)$$

Moreover, we have:

$$\begin{aligned} \sum_{i \in S} v_i \tilde{d}_i &= v_{[1]} \tilde{d}_{[1]} + \sum_{i \in S \setminus [1]} v_i \tilde{d}_i \\ \sum_{i \in S} v_i \tilde{d}_i &= v_{[1]} \tilde{d}_{[1]} + \sum_{i \in S \setminus [1]} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} + \sum_{i \in S \setminus [1]} \min \left\{ f_i + v_i \tilde{d}_i, \lambda_S \right\} \end{aligned}$$

Using the expression (38), we have:

$$\begin{aligned}
\sum_{i \in S} v_i \tilde{d}_i &= v_{[1]} \tilde{d}_{[1]} + \sum_{i \in S \setminus [1]} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} + \\
&\quad \max_{0 \leq x_{i'} \leq \frac{c-f_{i'}}{v_{i'}}} \left\{ \varphi_S (c - (f_{i'} + v_{i'} x_{i'})) + \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} (v_{i'} x_{i'} - (\mu_1 - f_{i'})) \right\} \\
\sum_{i \in S} v_i \tilde{d}_i &\geq v_{[1]} \tilde{d}_{[1]} + \sum_{i \in S \setminus [1]} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} + \\
&\quad \varphi_S (c - (f_{i'} + v_{i'} x_{i'}^*)) + \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} (v_{i'} x_{i'}^* - (\mu_1 - f_{i'})) \\
\sum_{i \in S} v_i \tilde{d}_i &\geq v_{[1]} \tilde{d}_{[1]} - \lambda_S + \lambda_S + \sum_{i \in S \setminus [1]} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} + \\
&\quad \varphi_S (c - (f_{i'} + v_{i'} x_{i'}^*)) + \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} (v_{i'} x_{i'}^* - (\mu_1 - f_{i'})) \tag{39}
\end{aligned}$$

Since  $v_{[1]} \tilde{d}_{[1]} - \lambda_S \geq -f_{[1]}$  (we know that :  $f_{[1]} + v_{[1]} \tilde{d}_{[1]} \geq \lambda_S$ ),  $v_{[1]} \tilde{d}_{[1]} - \lambda_S$  can be replaced by  $\max \left\{ -f_{[1]}, v_{[1]} \tilde{d}_{[1]} - \lambda_S \right\}$ . By rewriting the expression (39), we get (37). Then, we derive the inequality (33).

If  $|\bar{T}'| > 1$ , then the expression (33) can be easily generalized by considering the items which belong to  $\bar{T}'$  one by one. We get then the inequality (32).  $\square$

In what follows, we describe another class of valid inequalities based on the reverse cover set.

**Definition 2.** (*Reverse Cover*)

A subset  $S$  of  $\mathcal{I}$  is known as reverse cover of SPMCLSSP if:

$$\mu_S = c - \sum_{i \in S} (f_i + v_i \tilde{d}_i) \geq 0 \tag{40}$$

For a reverse cover  $S$ ,  $\mu_S$  expresses the available capacity left when the total requirement for each item of  $S$  is produced.

**Proposition 6.** Let  $S$  be a reverse cover of SPMCLSSP,  $T = \mathcal{I} \setminus S$  and  $(T', T'')$  be any partition of  $T$ . The inequality

$$\sum_{i \in S} v_i (\tilde{s}_i + \tilde{r}_i) \geq \left( \sum_{i \in S} (f_i + v_i \tilde{d}_i) \right) \sum_{i \in T'} y_i - \sum_{i \in S} f_i (1 - y_i) - \sum_{i \in T'} ((c - f_i) y_i - v_i x_i) \tag{41}$$

is valid for SPMCLSSP.

*Proof.* The proof presented here is similar to the one described in Miller *et al.* [26]. In the following, we take into account the demand shortage variables  $\tilde{r}_i$  as well as the variable resource consumption  $v_i$ .

Let  $(x^*, y^*, \tilde{s}^*, \tilde{r}^*)$  be any point of the convex hull of SPMCLSSP.

We have to consider three cases:

If  $y_i^* = 0$  for all  $i \in T'$ , then the inequality is valid, because  $\sum_{i \in S} v_i (\tilde{s}_i^* + \tilde{r}_i^*) \geq -\sum_{i \in S} f_i (1 - \tilde{y}_i^*)$ .

Let  $\bar{T}' = \{j \in T' : y_j^* = 1\}$

If  $|\bar{T}'| = 1$ , we assume that  $\bar{T}' = \{i'\}$

From (3) we have:

$$c - f_{i'} \geq \sum_{i \in S} (v_i x_i^* + f_i y_i^*) + v_{i'} x_{i'}^*$$

From (17) we also have:

$$x_i^* \geq \tilde{d}_i - \tilde{s}_i^* - \tilde{r}_i^*$$

Consequently, we get:

$$c - f_{i'} \geq \sum_{i \in S} \left( v_i (\tilde{d}_i - \tilde{s}_i^* - \tilde{r}_i^*) + f_i y_i^* \right) + v_{i'} x_{i'}^*$$

Which gives:

$$\sum_{i \in S} v_i (\tilde{s}_i^* + \tilde{r}_i^*) \geq \sum_{i \in S} v_i \tilde{d}_i + \sum_{i \in S} f_i y_i^* - ((c - f_{i'}) - v_{i'} x_{i'}^*)$$

The inequality

$$\sum_{i \in S} v_i (\tilde{s}_i^* + \tilde{r}_i^*) \geq \left( \sum_{i \in S} (f_i + v_i \tilde{d}_i) \right) - \sum_{i \in S} f_i (1 - y_i^*) - ((c - f_{i'}) - v_{i'} x_{i'}^*) \quad (42)$$

is thus valid for SPMCLSSP.

If  $|\bar{T}'| > 1$ , the inequality (42) can be easily generalized by considering the items of  $\bar{T}'$  one by one. The inequality (41) follows.  $\square$

## 6 Lifting cover and reverse cover inequalities

In this section, we will strengthen the valid inequalities by using superadditive functions for an iterative improvement. We refer the reader to [17] and [39] for a detailed description of lifting procedures using superadditive functions.

Our work is based on Marchand and Wolsey [24] work on the continuous knapsack problem, as well as the adaptations carried out by Miller *et al.* [26] for the problem  $(\bar{P}r)$  in order to lift cover and reverse cover inequalities for the SPMCLSSP problem.

### 6.1 The 0-1 continuous knapsack problem

Let us define the following problem:

$$Y = \left\{ (y, s) \in \{0, 1\}^n \times R_+^1 : \sum_{j \in \mathcal{J}} a_j y_j \leq b + s. \right\} \quad (43)$$

With:  $\mathcal{J} = \{1, \dots, n\}$ ,  $a_j \in \mathbb{Z}_+$ ,  $j \in \mathcal{J}$  and  $b \in \mathbb{Z}_+$ .

Let  $(\{j'\}, C, D)$  be a cover pair for  $Y$  such that:

- $C \cap D = \{j'\}$ ,  $C \cup D = \mathcal{J}$
- $\lambda_C = \sum_{j \in C} a_j - b > 0$
- $a_{j'} > \lambda_C$

We can notice that  $\mu_D = a_{j'} - \lambda_C = \sum_{j \in D} a_j - \left( \sum_{j \in \mathcal{J}} a_j - b \right) > 0$ .  $C$  is thus a cover set and  $D$  is a reverse cover set.

We will now recall main results on the cover and reverse cover inequalities defined for this problem.

**Proposition 7.** (*Continuous cover inequalities, Marchand and Wolsey [24]*)

Let  $(\{j'\}, C, D)$  be a cover pair for  $Y$ . We consider an order of the elements of  $C$  such that  $a_{[1]} \geq \dots \geq a_{[r_C]}$  where  $r_C$  is the number of elements of  $C$  with  $a_j > \lambda_C$ . Let us denote  $A_0 = 0$  and  $A_j = \sum_{p=1}^j a_{[p]}$ ,  $j = 1, \dots, r_C$ . We set:

$$\phi_C(u) = \begin{cases} (j-1)\lambda_C, & \text{if } A_{j-1} \leq u \leq A_j - \lambda_C, j = 1, \dots, r_C \\ (j-1)\lambda_C + [u - (A_j - \lambda_C)], & \text{if } A_{j-1} - \lambda_C \leq u \leq A_j, j = 1, \dots, r_C \\ (r_C - 1)\lambda_C + [u - (A_{r_C} - \lambda_C)], & \text{if } A_{r_C} - \lambda_C \leq u \end{cases} \quad (44)$$

The inequality

$$\sum_{j \in C} \min(\lambda_C, a_j) y_j + \sum_{D \setminus j'} \phi_C(a_j) y_j \leq \sum_{j \in C \setminus j'} \min(\lambda_C, a_j) + s \quad (45)$$

is valid for  $Y$  and defines a facet of  $\text{conv}(Y)$ .

A simple representation of  $\phi_C$  is given in Fig. 2.

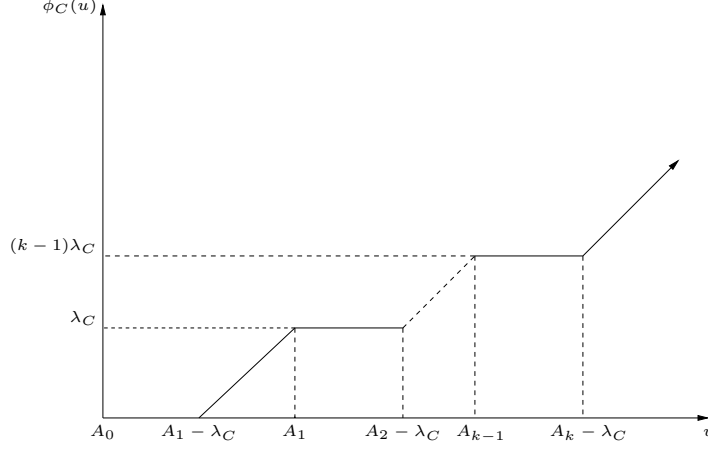


Figure 2: Function  $\phi_C$ .

**Proposition 8.** (Continuous reverse cover inequalities, Marchand and Wolsey [24])

Let  $(\{j'\}, C, D)$  be a cover pair for  $Y$ . We consider an order of the elements of  $D$  such that  $a_{[1]} \geq \dots \geq a_{[r_D]}$  where  $r_D$  is the number of elements of  $D$  with  $a_j > \mu_D$ , where  $\mu_D = a_{j'} - \lambda_C$ . Let  $A_0 = 0$  and  $A_j = \sum_{p=1}^j a_{[p]}$ ,  $j = 1, \dots, r_D$ . We set:

$$\psi_D(u) = \begin{cases} u - j\mu_D, & \text{if } A_j \leq u \leq A_{j+1} - \mu_D, j = 0, \dots, r_D - 1 \\ A_j - j\mu_D, & \text{if } A_j - \mu_D \leq u \leq A_j, j = 1, \dots, r_D - 1 \\ A_{r_D} - r_D\mu_D, & \text{if } A_{r_D} - \mu_D \leq u \end{cases} \quad (46)$$

The inequality

$$\sum_{j \in D} (a_j - \mu_D) y_j + \sum_{j \in C \setminus j'} \psi_D(a_j) y_j \leq \sum_{j \in C \setminus j'} \psi_D(a_j) + s \quad (47)$$

is valid for  $Y$  and defines a facet of  $\text{conv}(Y)$ .

A simple representation of  $\psi_C$  is given in Fig. 3.

## 6.2 Lifting cover inequalities for the SPMCLSSP problem

In what follows, we use the results of Marchand and Wolsey [24] to obtain valid inequalities stronger than (26). Let us recall that:

- $S$  is a cover for the SPMCLSSP problem.
- $T = \mathcal{I} \setminus S$ .
- $T', T''$  is a partition of  $T$ .
- $U \subset T''$ .

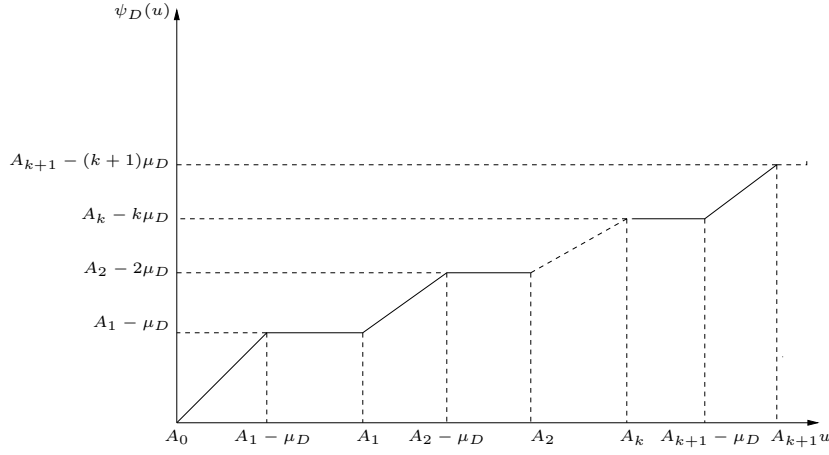


Figure 3: Function  $\psi_D$ .

From the constraints (3) and (17), we can write:

$$\sum_{i \in S \cup U} \left( f_i y_i + v_i \tilde{d}_i - v_i \tilde{s}_i - v_i \tilde{r}_i \right) \leq c$$

By adding  $\sum_{i \in S \cup U} v_i \tilde{d}_i y_i$  to both sides of this inequality, we see that:

$$\sum_{i \in S \cup U} v_i \tilde{d}_i y_i + \sum_{i \in S \cup U} \left( f_i y_i + v_i \tilde{d}_i - v_i \tilde{s}_i - v_i \tilde{r}_i \right) \leq c + \sum_{i \in S \cup U} v_i \tilde{d}_i y_i$$

Thus:

$$\sum_{i \in S \cup U} \left( f_i + v_i \tilde{d}_i \right) y_i \leq c + \sum_{i \in S \cup U} \left( v_i \tilde{s}_i + v_i \tilde{r}_i + v_i \tilde{d}_i y_i - v_i \tilde{d}_i \right) \quad (48)$$

If we denote  $u = \sum_{i \in S \cup U} \left( v_i \tilde{s}_i + v_i \tilde{r}_i + v_i \tilde{d}_i y_i - v_i \tilde{d}_i \right)$ , it results that:

$$\sum_{i \in S \cup U} \left( f_i + v_i \tilde{d}_i \right) y_i \leq c + u \quad (49)$$

The inequality (49) can thus be considered as a constraint of a 0-1 continuous knapsack problem. So, we get the following properties.

**Proposition 9.** *Given  $S$  a cover of SPMCLSSP, and  $U$  a subset of  $\mathcal{I} \setminus S$ , the inequality (where  $\phi_S$  is defined by (44))*

$$\sum_{i \in S \cup U} v_i (\tilde{s}_i + \tilde{r}_i) \geq \lambda_S + \sum_{i \in S} \max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} (1 - y_i) + \sum_{i \in U} \left( \phi_S \left( f_i + v_i \tilde{d}_i \right) \right) y_i + \sum_{i \in U} v_i \tilde{d}_i (1 - y_i) \quad (50)$$

*is valid for SPMCLSSP.*

*Proof.* Let  $(\{j'\}, U \cup \{j'\}, S)$  be a cover pair for SPMCLSSP such that:  $f_j + v_j \tilde{d}_j > \lambda_S$ . According to the proposition 7, the following inequality is valid for SPMCLSSP

$$\begin{aligned} \sum_{i \in S} \min \left( \lambda_S, f_i + v_i \tilde{d}_i \right) y_i + \sum_{i \in U} \phi_S \left( f_i + v_i \tilde{d}_i \right) y_i &\leq \sum_{i \in S \setminus \{j\}} \min \left( \lambda_S, f_i + v_i \tilde{d}_i \right) \\ &+ \sum_{i \in S \cup U} \left( v_i \tilde{s}_i + v_i \tilde{r}_i + v_i \tilde{d}_i y_i - v_i \tilde{d}_i \right) \end{aligned}$$

Since  $f_j + v_j \tilde{d}_j > \lambda_S$ , we have  $\min(f_j + v_j \tilde{d}_j, \lambda_S) = \lambda_S$ . By adding  $\min(f_j + v_j \tilde{d}_j, \lambda_S) - \lambda_S$  to the right side of the previous inequality, we get:

$$\begin{aligned} \sum_{i \in S} \min(\lambda_S, f_i + v_i \tilde{d}_i) y_i + \sum_{i \in U} \phi_S(f_i + v_i \tilde{d}_i) y_i &\leq \sum_{i \in S} \min(\lambda_S, f_i + v_i \tilde{d}_i) - \lambda_S \\ &\quad + \sum_{i \in S \cup U} (v_i \tilde{s}_i + v_i \tilde{r}_i + v_i \tilde{d}_i y_i - v_i \tilde{d}_i) \end{aligned}$$

which is valid for SPMCLSSP. We obtain the inequation (50) by simplification.  $\square$

**Proposition 10.** *Let  $S$  be a cover of SPMCLSSP. We consider an order  $[1], \dots, [|S|]$  such that  $f_{[1]} + v_{[1]} \tilde{d}_{[1]} \geq \dots \geq f_{[|S|]} + v_{[|S|]} \tilde{d}_{[|S|]}$ . Let us set  $T = \mathcal{I} \setminus S$  and  $(T', T'')$  any partition of  $T$ . We define  $\mu_1 = f_{[1]} + v_{[1]} \tilde{d}_{[1]} - \lambda_S$ .*

*If  $|S| \geq 2$  and  $f_{[2]} + v_{[2]} \tilde{d}_{[2]} \geq \lambda_S$ , the inequality*

$$\begin{aligned} \sum_{i \in S \cup U} v_i (\tilde{s}_i + \tilde{r}_i) &\geq \lambda_S + \sum_{i \in S} \max\{-f_i, v_i \tilde{d}_i - \lambda_S\} (1 - y_i) + \sum_{i \in U} \phi_S(f_i + v_i \tilde{d}_i) y_i \\ &\quad + \sum_{i \in U} v_i \tilde{d}_i (1 - y_i) + \frac{\lambda_S}{f_{[2]} + v_{[2]} \tilde{d}_{[2]}} \sum_{i \in T'} (v_i x_i - (\mu_1 - f_i) y_i) \end{aligned} \quad (51)$$

*is valid for SPMCLSSP.*

*Proof.* The proof is similar to the proof of proposition 5.  $\square$

### 6.3 Lifting reverse cover inequalities for the SPMCLSSP problem

In the same way, we can notice that the inequality (48) is a constraint of a 0-1 continuous knapsack problem. The following propositions hold.

**Proposition 11.** *Let  $S$  be a reverse cover for SPMCLSSP and  $U$  a subset of  $\mathcal{I} \setminus S$ . The inequality*

$$\sum_{i \in S \cup U} v_i (\tilde{s}_i + \tilde{r}_i) \geq \sum_{i \in S} (v_i \tilde{d}_i - \psi_U(f_i + v_i \tilde{d}_i)) (1 - y_i) + \sum_{i \in U} v_i \tilde{d}_i + \sum_{i \in U} (f_i - \mu_S) y_i \quad (52)$$

*is valid for SPMCLSSP.*

*Proof.* According to proposition 8, the inequality

$$\sum_{i \in S} \psi_U(f_i + v_i \tilde{d}_i) y_i + \sum_{i \in U} (f_i + v_i \tilde{d}_i - \mu_S) y_i \leq \sum_{i \in S} \psi_U(f_i + v_i \tilde{d}_i) + \sum_{i \in S \cup U} v_i (\tilde{s}_i + \tilde{r}_i + \tilde{d}_i y_i - \tilde{d}_i)$$

is valid for SPMCLSSP. We obtain the inequation (52) by simplification.  $\square$

**Proposition 12.** *Let  $S$  be a reverse cover of the SPMCLSSP problem and  $U \subset \mathcal{I} \setminus S$  such that  $f_i + v_i \tilde{d}_i \geq \mu_S \ \forall i \in U$ . We consider an order  $[1], \dots, [|U|]$  such that  $f_{[1]} + v_{[1]} \tilde{d}_{[1]} \geq \dots \geq f_{[|U|]} + v_{[|U|]} \tilde{d}_{[|U|]}$ . Let  $T = \mathcal{I} \setminus \{S \cup U\}$  and  $(T', T'')$  any partition of  $T$ . The inequality  $(\psi_U$  is defined by (46)):*

$$\begin{aligned} \sum_{i \in S \cup U} v_i (\tilde{s}_i + \tilde{r}_i) &\geq \sum_{i \in S} (v_i \tilde{d}_i - \psi_U(f_i + v_i \tilde{d}_i)) (1 - y_i) + \sum_{i \in U} (f_i - \mu_S) y_i + \sum_{i \in U} v_i \tilde{d}_i \\ &\quad + \min_{i \in S} \left\{ \frac{\psi_U(f_i + v_i \tilde{d}_i)}{f_i + v_i \tilde{d}_i} \right\} \sum_{i \in T'} (v_i x_i - (\mu_S - f_i) y_i) \end{aligned} \quad (53)$$

*is valid for SPMCLSSP.*

*Proof.* The proof of this proposition is similar to one described for proposition 4.  $\square$



## 7 Separation heuristic for cover inequalities

In this section, we present a fast combinatorial separation heuristic to create cover inequalities for the SPMCLSSP problem, which are also valid for the MCLSSP. Indeed, for the latter problem, we generate the cover inequalities for each period of the planning horizon which corresponds to the cover inequalities of the SPMCLSSP.

In order to build a cover inequality for the SPMCLSSP problem, the first step is to define a cover set  $S$ , then we compute  $\lambda_S$  and  $\mu_1$  (see (25) and proposition 5). The second step is to examine all the elements  $i \in \mathcal{I} \setminus S$  to create the sets  $U$  and  $T'$ .

We use a greedy algorithm to create the set  $S$ . We sort the elements  $i \in \mathcal{I}$  according to the descending order of the value:

$$\max \left\{ -f_i, v_i \tilde{d}_i - \lambda_S \right\} (1 - y_i^*) - v_i (\tilde{s}_i^* + \tilde{r}_i^*) \quad (54)$$

The formula (54) is obtained from the inequality (51) by considering the only terms in relation with  $S$ . The value of (54) represents the contribution of the violation of the inequality (51) by the item  $i \in S$  and depends on  $\lambda_S$ . However,  $\lambda_S$  is not known in advance. Therefore, we estimate the value of  $\lambda_S$ . To do that, we sort the items  $i \in \mathcal{I}$  according to the descending order of their resource consumption by using the formula (55). Formula (54) would give a better set  $S$  but cannot be used since  $\lambda_S$  is not known, so in practice formula (55) is used.

$$(f_i + v_i \tilde{d}_i) y_i^* \quad (55)$$

We can notice that the formula (55) represents the resource consumption of the item  $i$  if the total requirement  $\tilde{d}_i$  is produced.

In order to create a cover set  $S$ , we greedily add the sorted elements according to the formula (55) until we get a cover set. For the design of the set  $U$  (respectively  $T'$ ), we examine all the elements  $i \in \mathcal{I} \setminus S$  and check if the corresponding value of the expression obtained by summing up the terms of the inequality (51) in relation with  $U$  (respectively to  $T'$ ) is positive. In this case, we add the elements  $i$  to  $U$  (respectively to  $T'$ ). We derive a valid inequality using (51) with the sets  $S$ ,  $U$  and  $T'$  obtained. If the value of the inequality is positive, we get a cut.

The basic principle previously described is captured in the following algorithm.

### Algorithm 2 Separation heuristic for cover inequalities

- 1: Order the elements of  $\mathcal{I}$  in descending order according to the formula (55).
- 2:  $S \leftarrow \emptyset, U \leftarrow \emptyset, T' \leftarrow \emptyset, i' \leftarrow 0$
- 3:  $i' \leftarrow \arg \min_{i=1, \dots, N} \left\{ \sum_{k=1}^i (f_{[k]} + v_{[k]} \tilde{d}_{[k]}) > c \right\}$
- 4:  $S \leftarrow \{[1], \dots, [i']\}$
- 5:  $i_1 \leftarrow \arg \max_{i \in S} \{f_i + v_i \tilde{d}_i\}$
- 6:  $i_2 \leftarrow \arg \max_{i \in S \setminus \{i_1\}} \{f_i + v_i \tilde{d}_i\}$
- 7:  $\lambda_S \leftarrow \left( \sum_{k=1}^{i'} f_{[k]} + v_{[k]} \tilde{d}_{[k]} \right) - c$
- 8: **if**  $(|S| \geq 2)$  and  $(f_{i_2} + v_{i_2} \tilde{d}_{i_2} \geq \lambda_S)$  **then**
- 9:    $\mu_1 \leftarrow \{f_{i_1} + v_{i_1} \tilde{d}_{i_1}\} - \lambda_S$
- 10:    $i \leftarrow i' + 1$
- 11: **while**  $(i \leq N)$  **do**
- 12:   **if**  $(\phi_S (f_{[i]} + v_{[i]} \tilde{d}_{[i]}) y_{[i]}^* + v_{[i]} \tilde{d}_{[i]} (1 - y_{[i]}^*) - v_{[i]} (\tilde{s}_{[i]}^* + \tilde{r}_{[i]}^*) > 0)$  **then**
- 13:      $U \leftarrow U \cup \{[i]\}$
- 14:   **else if**  $(v_{[i]} x_{[i]}^* > (\mu_1 - f_{[i]}) y_{[i]}^*)$  **then**
- 15:      $T' \leftarrow T' \cup \{[i]\}$

```

16:   end if
17:    $i \leftarrow i + 1$ 
18: end while
19: end if
20: if (The inequality (51) based on  $S, U$  and  $T'$  is violated) then
21:   Add the inequality (51) at the current node.
22: end if

```

We recall that the evaluation of the superadditive functions  $\phi_S$  is in  $O(N)$  for each item. The separation heuristic for generating the cover inequalities for the SPMCLSSP problem is obviously in  $O(N^2)$ . Moreover, since each period is examined separately to create valid inequalities for the MCLSSP problem, the separation heuristic is then in  $O(N^2T)$  for the latter problem.

## 8 Separation heuristic for reverse cover inequalities

The idea of the separation heuristic for reverse cover inequalities is similar to the previous one. The first step is to create a reverse cover set  $S$  in order to define  $\mu_S$  (see (40)). The second step is to examine all the elements  $i \in \mathcal{I} \setminus S$  to create the sets  $U$  and  $T'$ . We use a greedy algorithm to create  $S$  by sorting the elements  $i \in \mathcal{I}$  according to the descending order of the value:

$$\left( v_i \tilde{d}_i - \psi_U \left( f_i + v_i \tilde{d}_i \right) \right) (1 - y_i^*) - v_i (\tilde{s}_i^* + \tilde{r}_i^*) \quad (56)$$

In the same way, the formula (56) is obtained from the inequality (53) by considering the only terms in relation with  $S$ . The value of (56) represents the contribution of the violation of the inequality (53) by the item  $i \in S$  and depends on  $\mu_S$ . However,  $\mu_S$  is not known in advance. Therefore, we estimate the value of  $\mu_S$ . To do that, we sort the items  $i \in \mathcal{I}$  according to the descending order of their resource consumption by using the formula (55). Formula (56) would give a better set  $S$  but cannot be used since  $\lambda_S$  is not known, so in practice formula (55) is used. We illustrate this principle in the following algorithm:

**Algorithm 3** Separation heuristic for reverse cover inequalities

```

1: Order the elements of  $\mathcal{I}$  in descending order according to the formula (55)
2:  $S \leftarrow \emptyset, i' \leftarrow 1$ 
3: while ( $i' \leq N$ ) do
4:    $S \leftarrow S \cup \{i'\}$ 
5:    $\mu_S = c - \sum_{j=1}^{i'} f_{[j]} + v_{[j]} \tilde{d}_{[j]}$ 
6:   if ( $\mu_S < 0$ ) then
7:      $T' \leftarrow \emptyset, U \leftarrow \emptyset, i \leftarrow i' + 1$ 
8:     while ( $i \leq N$ ) do
9:       if ( $(f_{[i]} - \mu_S) y_{[i]}^* + v_{[i]} \tilde{d}_{[i]} - v_{[i]} (\tilde{s}_{[i]}^* + \tilde{r}_{[i]}^*) > 0$ ) then
10:         $U \leftarrow U \cup \{[i]\}$ 
11:       else if ( $(v_{[i]} x_{[i]}^* - (\mu_S - f_{[i]}) y_{[i]}^*) > 0$ ) then
12:         $T' \leftarrow T' \cup \{[i]\}$ 
13:       end if
14:        $i \leftarrow i + 1$ 
15:     end while
16:     if (The inequality (53) based on  $S, U$  and  $T'$  is violated) then
17:       Add the inequality (53) at the current node.
18:        $i' \leftarrow N + 1$ 
19:     else
20:        $i' \leftarrow i' + 1$ 
21:     end if
22:   else
23:      $i' \leftarrow N + 1$ 

```

24: **end if**  
25: **end while**

We recall that the evaluation of the superadditive functions  $\psi_S$  is in  $O(N)$  for each item. The separation heuristic for constructing reverse cover inequalities for the SPMCLSSP problem is obviously in  $O(N^2)$ . Since each period is examined separately to create valid inequalities for the MCLSSP problem, the separation heuristic is then in  $O(N^2T)$  for the MCLSSP.

## 9 Computational issues and results

In this section, we discuss computational issues that arise in using the classes of inequalities previously identified. We report computational results from cut-and-branch and branch-and-cut frameworks.

The cut-and-branch method consists in adding cuts only at the first node (or root) of the branch-and-bound tree in order to improve the lower bound. The branch-and-cut method consists in adding cuts not only at the first node but at other nodes of the branch-and-bound tree. Usually, cuts are not added at all the nodes of the branch-and-bound tree in order not to slow down the total CPU time while solving the problem.

Our algorithm is implemented in the C++ programming language and it is integrated in an APS software. It uses the callable CPLEX 9.0 library [20] that provides callback functions that allow the user to implement his own branch-and-cut algorithm.

We have performed computational tests on a series of extended instances from the lot-sizing library LOTSIZELIB [22], initially described in Trigeiro *et al.* [36] and also used by Miller [25]. Trigeiro *et al.* [36] instances are denoted  $tr_{N-T}$ , where  $N$  is the number of items and  $T$  is the number of periods. These are characterized by a variable resource consumption equal to one, and enough capacity to satisfy all the requirement over the planning horizon. They are also characterized, by an important setup cost, a small fixed resource requirement (setup time) and no  $\sigma_{it}$  which denotes the last period at which an item  $i$  produced at period  $t$  can be consumed.

The characteristics of Trigeiro *et al.* [36] problems are presented in table 1.

| Instance     | N  | T  |
|--------------|----|----|
| $tr_{6-15}$  | 6  | 15 |
| $tr_{6-30}$  | 6  | 30 |
| $tr_{12-15}$ | 12 | 15 |
| $tr_{12-30}$ | 12 | 30 |
| $tr_{24-15}$ | 24 | 15 |
| $tr_{24-30}$ | 24 | 30 |

Table 1: Instances of Trigeiro *et al.* [36].

Since these instances have enough capacity to satisfy all the requirements over the planning horizon, we make some modifications to induce shortages. We have derived 24 new benchmarks<sup>2</sup> from the  $tr_{N-T}$  instances by augmenting the fixed resource requirements (setup times), the variable resource requirements and by adding  $\sigma_{it}$ . We have also generated shortage costs. More details are given below.

These new benchmarks fall into 4 classes of 6 instances each:

- The first class was obtained by increasing the variable resource requirements and adding  $\sigma_{it}$ . Variable resource requirements are multiplied by a coefficient  $(1 + \rho)$  such that  $0 \leq \rho \leq 0.001 \times c_t$ ,  $c_t$  represents the available resource capacity at period  $t$ .  $\sigma_{it}$  are generated such that we cannot anticipate production more than  $\frac{1}{3}T$  periods,  $T$  denotes the number of periods.

---

<sup>2</sup>Test problems can be obtained from the corresponding author.

- The second class is obtained by carrying out the same modifications on the variable resource requirements than the first class.  $\sigma_{it}$  are generated such that we cannot anticipate production more than  $\frac{2}{3}T$  periods.
- The third class is based on the first one. In fact, we carried out some modifications on fixed resource requirements which are increased by multiplying them by a coefficient  $(1 + \tau)$  such that  $\tau \approx 0.1 \times c_t$ .
- The last class is obtained by carrying out the same modifications on the variable and fixed resource requirements than the third class.  $\sigma_{it}$  are generated such that we cannot anticipate production more than  $\frac{2}{3}T$  periods.

Shortage costs are considered as penalty costs and their values must be higher than other cost components. Therefore,  $\varphi_{it}$  are fixed such that  $\varphi_{it} \gg \max_{i',t'} \{\alpha_{i't'}; \beta_{i't'}; \gamma_{i't'}\}$ . Moreover, shortage costs have the feature that they decrease over the horizon. In fact, demands in the first periods of the horizon correspond to real orders and not forecasts by opposition to the demands in the last periods that are usually only predictions. They are generated in the same way for all the described instances.

We carried out a comparison between the following methods:

- An algorithm based on the standard branch-and-cut of CPLEX solver that we denote by BC.
- An algorithm based on the standard branch-and-cut of CPLEX solver including all the cuts presented in this paper denoted by BC+.

In both algorithms BC and BC+, we used the aggregated model defined in section 1 by the set of constraints (1)-(7).

Two kinds of specialized cuts in mixed linear problems are used in both methods, the flow cover cuts and the Mixed Integer Rounding (MIR) cuts from CPLEX 9.0 solver. The flow cover cuts are accurate regarding the MCLSSP problem because of the flow structure induced by the flow conservation constraints (2). For a complete description of these flow cover cuts, the reader can refer to Gomory [16], Nemhauser and Wolsey [27] and Wolsey [40]. The MIR cuts were primarily applied to both capacity and maximum production constraints. For more details on the MIR cuts, we can refer to Padberg *et al.* [30] and Van Roy and Wolsey [34].

For all the algorithms,  $LB$  and  $UB$  represent respectively the lower bound and the upper bound values at the termination of the algorithm.  $NB_{Nodes}$  is the number of the nodes explored in the branch-and-bound tree,  $U_{Cuts}$  is the number of the cuts added during the branch-and-cut algorithm (cover, reverse cover and  $(l, S)$  inequalities).  $F_{Cuts}$  and  $MIR_{Cuts}$  represents respectively the number of flow cover and MIR cuts added by the solver during the branch-and-cut algorithm. All the algorithm comparisons are based on the following criteria. The first one called GAP is equal to  $|UB - LB| / |UB|$ , and the second one is a CPU time denoted  $Time$ . The computations are performed on a Pentium IV 2.66 Ghz PC.

At the root node of the BC+ method, we use algorithms 1, 2 and 3 (see sections 4, 7 and 8) until we do not find any more violated inequalities. The same procedure is followed in the branch-and-bound tree.

The branching strategy in both algorithms is depth-first search to find a feasible solution. Upper bounds are either obtained when LP solutions are integral or by the LP based heuristics of the solver.

Generally, our computational results show that adding inequalities at the root node improves considerably the lower bounds. The average improvement of the lower bound at the root node of BC+ is 80% for the first class, 53% for the second class, 73% for the third class and 48% for the last class. This rate is the percentage obtained between the best lower bound observed at the first node of BC and the best one found at the end of the BC+ method.

Table 2 summarizes the computational behaviour based on a time-limit criterion. We allow a maximum of 600 seconds CPU time for all the algorithms.

From table 2, we can easily notice that using the valid inequalities described in this paper improves the performance of the branch-and-cut algorithm. Clearly BC+ solves the test problems more effectively than BC. The valid inequalities that we have proposed are interesting since all the lower bounds given by BC+ are better than those given by BC.

| $N$     | $T$ | Method | $UB$       | $LB$       | $NB_{Nodes}$ | $U_{Cuts}$ | $MIRCuts$ | $F_{Cuts}$ | $GAP$  |
|---------|-----|--------|------------|------------|--------------|------------|-----------|------------|--------|
| Class 1 |     |        |            |            |              |            |           |            |        |
| 6       | 15  | BC     | 4 038 212  | 3 979 268  | 123 900      | 0          | 309       | 254        | 1,46%  |
| 6       | 15  | BC+    | 4 030 456  | 3 999 352  | 70 000       | 397        | 87        | 160        | 0,77%  |
| 6       | 30  | BC     | 4 536 980  | 4 124 370  | 34 000       | 0          | 625       | 523        | 9,09%  |
| 6       | 30  | BC+    | 4 392 289  | 4 271 466  | 10 300       | 877        | 100       | 200        | 2,75%  |
| 12      | 15  | BC     | 7 669 660  | 7 495 023  | 62 600       | 0          | 616       | 464        | 2,28%  |
| 12      | 15  | BC+    | 7 651 166  | 7 610 635  | 19 600       | 517        | 257       | 300        | 0,53%  |
| 12      | 30  | BC     | 8 772 505  | 7 903 312  | 25 600       | 0          | 607       | 717        | 9,91%  |
| 12      | 30  | BC+    | 8 609 716  | 8 345 637  | 1 500        | 2 063      | 64        | 364        | 3,07%  |
| 24      | 15  | BC     | 14 117 000 | 13 780 000 | 73 600       | 0          | 370       | 600        | 2,39%  |
| 24      | 15  | BC+    | 14 082 000 | 13 995 000 | 10 300       | 796        | 83        | 445        | 0,62%  |
| 24      | 30  | BC     | 23 619 000 | 22 879 000 | 11 800       | 0          | 837       | 1 298      | 3,13%  |
| 24      | 30  | BC+    | 23 558 000 | 23 180 000 | 300          | 4 048      | 134       | 807        | 1,60%  |
| Class 2 |     |        |            |            |              |            |           |            |        |
| 6       | 15  | BC     | 4 032 790  | 3 978 178  | 120 900      | 0          | 304       | 253        | 1,35%  |
| 6       | 15  | BC+    | 4 034 967  | 3 987 919  | 47 400       | 248        | 157       | 2 001      | 1,17%  |
| 6       | 30  | BC     | 4 530 576  | 4 120 298  | 34 700       | 0          | 637       | 499        | 9,06%  |
| 6       | 30  | BC+    | 4 520 148  | 4 269 236  | 6 800        | 667        | 217       | 274        | 5,55%  |
| 12      | 15  | BC     | 7 664 197  | 7 477 993  | 57 600       | 0          | 546       | 532        | 2,43%  |
| 12      | 15  | BC+    | 7 666 557  | 7 528 738  | 14 500       | 439        | 201       | 393        | 1,80%  |
| 12      | 30  | BC     | 8 793 271  | 7 899 450  | 22 800       | 0          | 630       | 693        | 10,16% |
| 12      | 30  | BC+    | 8 675 000  | 8 250 948  | 1 900        | 1 506      | 240       | 456        | 4,89%  |
| 24      | 15  | BC     | 14 118 000 | 13 776 000 | 69 600       | 0          | 315       | 669        | 2,43%  |
| 24      | 15  | BC+    | 14 112 000 | 13 850 000 | 6 800        | 731        | 133       | 619        | 1,86%  |
| 24      | 30  | BC     | 23 634 000 | 22 875 000 | 13 200       | 0          | 742       | 1 335      | 3,21%  |
| 24      | 30  | BC+    | 23 647 000 | 23 002 000 | 300          | 2 798      | 388       | 1 120      | 2,73%  |
| Class 3 |     |        |            |            |              |            |           |            |        |
| 6       | 15  | BC     | 5 300 603  | 5 167 025  | 74 000       | 0          | 347       | 216        | 2,52%  |
| 6       | 15  | BC+    | 5 268 911  | 5 228 874  | 53 000       | 375        | 205       | 129        | 0,76%  |
| 6       | 30  | BC     | 7 210 627  | 6 581 190  | 34 400       | 0          | 664       | 484        | 8,73%  |
| 6       | 30  | BC+    | 7 138 995  | 6 798 534  | 14 000       | 539        | 372       | 251        | 4,77%  |
| 12      | 15  | BC     | 12 345 000 | 11 632 000 | 5 700        | 0          | 692       | 358        | 5,77%  |
| 12      | 15  | BC+    | 12 178 000 | 11 945 000 | 5 900        | 1 239      | 179       | 191        | 1,91%  |
| 12      | 30  | BC     | 15 728 000 | 12 710 000 | 14 500       | 0          | 1 014     | 456        | 19,19% |
| 12      | 30  | BC+    | 15 641 000 | 14 681 000 | 3 700        | 2 160      | 74        | 153        | 6,14%  |
| 24      | 15  | BC     | 26 330 000 | 24 822 000 | 10 700       | 0          | 850       | 445        | 5,73%  |
| 24      | 15  | BC+    | 26 038 000 | 25 613 000 | 3 500        | 603        | 437       | 408        | 1,63%  |
| 24      | 30  | BC     | 43 446 000 | 36 863 000 | 3 600        | 0          | 941       | 993        | 15,15% |
| 24      | 30  | BC+    | 43 476 000 | 41 702 000 | 10           | 3 559      | 55        | 292        | 4,08%  |
| Class 4 |     |        |            |            |              |            |           |            |        |
| 6       | 15  | BC     | 5 295 547  | 5 170 187  | 79 200       | 0          | 324       | 230        | 2,37%  |
| 6       | 15  | BC+    | 5 312 724  | 5 204 658  | 48 710       | 25 386     | 190       | 160        | 2,03%  |
| 6       | 30  | BC     | 7 202 452  | 6 581 498  | 32 800       | 0          | 674       | 462        | 8,62%  |
| 6       | 30  | BC+    | 7 059 530  | 6 736 455  | 11 600       | 14 323     | 414       | 277        | 4,58%  |
| 12      | 15  | BC     | 12 256 000 | 11 222 000 | 21 400       | 0          | 775       | 304        | 8,43%  |
| 12      | 15  | BC+    | 12 348 667 | 11 724 138 | 6 221        | 8 332      | 269       | 255        | 5,06%  |
| 12      | 30  | BC     | 15 697 000 | 12 520 000 | 18 000       | 0          | 932       | 492        | 20,24% |
| 12      | 30  | BC+    | 15 388 918 | 14 102 211 | 2 293        | 9 234      | 463       | 420        | 8,36%  |
| 24      | 15  | BC     | 26 463 000 | 23 996 000 | 14 500       | 0          | 798       | 466        | 9,32%  |
| 24      | 15  | BC+    | 26 452 668 | 25 207 181 | 3 180        | 6 268      | 427       | 380        | 4,71%  |
| 24      | 30  | BC     | 43 364 000 | 35 593 000 | 4 600        | 0          | 959       | 926        | 17,92% |
| 24      | 30  | BC+    | 42 319 192 | 38 745 933 | 160          | 3 313      | 460       | 604        | 8,44%  |

Table 2: Computational results: time-limit criterion.

We also can notice that the upper bounds obtained by  $BC+$  are better than those obtained by  $BC$ . In fact, usually a good lower bound implies a better upper bound. One reason is that a better lower bound can be used to prune dominated nodes of the research tree and allow the branch-and-bound algorithm to visit more interesting branches to find better solutions. An additional reason is that when the LP relaxation is tighter, it is easier to find integer solutions, either because the LP solution is more often integral or because LP based heuristics of the solver are more successful.

Moreover, the number of nodes explored by the  $BC$  method is much more higher than the one explored by  $BC+$ . The ratio between these two numbers varies from 150% to 500% for the small instances and 300% to 36000% for the bigger instances. In fact, generating cuts takes time and having many cuts slows down the LP resolution at each node. Thus, when we enable the cuts that we have developed, the solver generates a number of cuts lower than the number obtained when they are disabled. Since we generate cuts before the solver, the previous observation can be explained by the fact that these cuts dominate part of standard cuts generated by the solver.

Another remark that we can make by visualizing table 2 is that the third and forth classes are more difficult than the first and the second ones. Indeed, the third and forth class problems are characterized by a higher fixed resource consumption values than the first and the second ones.

We can also notice that problems with a small  $\sigma_{it}$  have a higher  $GAP$  than the ones with a big one. In fact, instances of class 1 and class 3 have reach a smaller  $GAP$  than respectively instances from class 2 and class 4 when using both algorithms  $BC$  and  $BC+$ .

Table 3 shows the percentages of generated cuts by  $BC+$  for all instances.  $\%lS_{cuts}$ ,  $\%C_{cuts}$  and  $\%RC_{cuts}$  are respectively the percentage of  $(l, S)$ , cover and reverse cover cuts generated by  $BC+$ . We use a time-limit criterion of 600 seconds for  $BC+$ .

| $N$     | $T$ | $\%lS_{cuts}$ | $\%C_{cuts}$ | $\%RC_{cuts}$ | $N$     | $T$ | $\%lS_{cuts}$ | $\%C_{cuts}$ | $\%RC_{cuts}$ |
|---------|-----|---------------|--------------|---------------|---------|-----|---------------|--------------|---------------|
| Class 1 |     |               |              |               | Class 3 |     |               |              |               |
| 12      | 15  | 4,08%         | 95,84%       | 0,08%         | 12      | 15  | 21,48%        | 77,20%       | 1,32%         |
| 12      | 30  | 14,62%        | 85,32%       | 0,05%         | 12      | 30  | 33,93%        | 64,89%       | 1,17%         |
| 24      | 15  | 13,16%        | 86,80%       | 0,04%         | 24      | 15  | 51,19%        | 34,24%       | 14,57%        |
| 24      | 30  | 43,96%        | 55,93%       | 0,11%         | 24      | 30  | 71,64%        | 21,44%       | 6,92%         |
| 6       | 15  | 18,78%        | 81,16%       | 0,06%         | 6       | 15  | 71,00%        | 27,43%       | 1,56%         |
| 6       | 30  | 71,16%        | 28,80%       | 0,04%         | 6       | 30  | 87,49%        | 10,18%       | 2,34%         |
| Class 2 |     |               |              |               | Class 4 |     |               |              |               |
| 12      | 15  | 2,58%         | 97,35%       | 0,06%         | 12      | 15  | 9,30%         | 90,18%       | 0,52%         |
| 12      | 30  | 6,60%         | 93,30%       | 0,11%         | 12      | 30  | 15,69%        | 83,11%       | 1,20%         |
| 24      | 15  | 22,97%        | 77,03%       | 0,00%         | 24      | 15  | 45,11%        | 45,11%       | 9,78%         |
| 24      | 30  | 27,31%        | 72,68%       | 0,01%         | 24      | 30  | 59,13%        | 38,36%       | 2,51%         |
| 6       | 15  | 31,41%        | 68,56%       | 0,04%         | 6       | 15  | 69,78%        | 27,98%       | 2,23%         |
| 6       | 30  | 60,46%        | 39,49%       | 0,05%         | 6       | 30  | 80,05%        | 17,05%       | 2,90%         |

Table 3: Computational results: cuts percentages.

From table 3, we can notice that the percentage of reverse cover cuts generated by  $BC+$  is very small. We can also notice that  $BC+$  generates more  $(l, S)$  cuts than cover cuts for class 1 and class 2 benchmarks except for instances with 24 items and 30 periods. For class 3 and class 4 instances,  $BC+$  generates more cover cuts than  $(l, S)$  cuts except for instances with 6 items.

We have also tested  $BC+$  using each family of valid inequalities separately ( $(l, S)$ , cover and reverse cover inequalities). Table 4 summarizes the computational results on class 1 and class 3 of instances.  $GAP_{lS}$ ,  $GAP_C$  and  $GAP_{RC}$  represents respectively the  $GAP$  when only the family of  $(l, S)$  inequalities is used, the  $GAP$  when only the family of cover inequalities is used and the  $GAP$  when only the family of reverse cover inequalities is used.

These tests show that the family of  $(l, S)$  inequalities is the most effective. The family of cover inequalities is less effective than the family of  $(l, S)$  inequalities. Using the family of reverse cover

| $N$     | $T$ | $GAP_{IS}$ | $GAP_C$ | $GAP_{CR}$ |
|---------|-----|------------|---------|------------|
| Class 1 |     |            |         |            |
| 12      | 15  | 0,56%      | 2,45%   | 2,40%      |
| 12      | 30  | 4,93%      | 11,03%  | 10,02%     |
| 24      | 15  | 0,55%      | 2,50%   | 2,54%      |
| 24      | 30  | 1,87%      | 3,37%   | 3,24%      |
| 6       | 15  | 0,68%      | 1,40%   | 1,39%      |
| 6       | 30  | 5,33%      | 9,44%   | 9,10%      |
| Class 3 |     |            |         |            |
| 12      | 15  | 1,48%      | 3,16%   | 4,45%      |
| 12      | 30  | 11,17%     | 14,71%  | 19,40%     |
| 24      | 15  | 2,43%      | 4,28%   | 5,32%      |
| 24      | 30  | 10,65%     | 12,40%  | 14,63%     |
| 6       | 15  | 1,19%      | 2,20%   | 2,60%      |
| 6       | 30  | 4,88%      | 7,27%   | 7,71%      |

Table 4: Computational results by cuts family.

inequalities alone is not really effective.

To give a relevant comparison concerning the number of added cuts and explored nodes for both algorithms  $BC$  and  $BC+$ , we solve some problems to a given  $GAP$ . Since we cannot solve these problems to optimality in a reasonable CPU time, we use a time-limit criterion of 1800 seconds for  $BC$ . We use the  $GAP$  obtained at the end of  $BC$  as a stopping criterion for  $BC+$ . We also use a time-limit of 1800 seconds for  $BC+$ . Table 5 summarizes the computational behaviour of the instances of class 1 based on a minimum  $GAP$  criterion.

| $N$     | $T$ | Method | $NB_{Nodes}$ | $U_{Cuts}$ | $MIR_{Cuts}$ | $F_{Cuts}$ | $GAP$ | $Time$ |
|---------|-----|--------|--------------|------------|--------------|------------|-------|--------|
| Class 1 |     |        |              |            |              |            |       |        |
| 6       | 15  | BC     | 420 821      | 0          | 309          | 254        | 1,03% | 1800   |
| 6       | 15  | BC+    | 9 810        | 517        | 14           | 183        | 0,94% | 53     |
| 6       | 30  | BC     | 105 911      | 0          | 625          | 523        | 8,51% | 1800   |
| 6       | 30  | BC+    | 110          | 1924       | 64           | 364        | 7,65% | 13     |
| 12      | 15  | BC     | 174 601      | 0          | 617          | 464        | 2,18% | 1800   |
| 12      | 15  | BC+    | 40           | 783        | 33           | 377        | 2,05% | 2      |
| 12      | 30  | BC     | 75 836       | 0          | 747          | 808        | 9,60% | 1800   |
| 12      | 30  | BC+    | 0            | 4048       | 134          | 807        | 8,66% | 35     |
| 24      | 15  | BC     | 206 400      | 0          | 433          | 638        | 2,14% | 1800   |
| 24      | 15  | BC+    | 0            | 326        | 88           | 158        | 0,88% | 4      |
| 24      | 30  | BC     | 42 796       | 0          | 880          | 1 417      | 3,00% | 1800   |
| 24      | 30  | BC+    | 10           | 848        | 81           | 184        | 2,00% | 175    |

Table 5: Computational results: minimum  $GAP$  criterion.

From table 5, we can easily notice that to reach the same  $GAP$ ,  $BC+$  does not need a significant number of nodes and time comparing to  $BC$ . We remark that without branching,  $BC+$  gets a lower  $GAP$  than  $BC$ 's one for the instances with 12 items and 30 periods and the one with 30 items and 24 periods. The cuts improve considerably the  $GAP$  at the root node.

Many authors reported experimental results showing that using the facility location-based formulation provide a better LP relaxation based lower bound than the one obtained by the aggregated formulation. (see [8], [31]). We carried out some computational experiments using the



facility location-based formulation introduced initially by Krarup and Bilde [18]. Production and stock variables are redefined by considering the period where the production is really consumed. These are reformulated using respectively formula (10) and (11) (see page 4).

We denote  $BC_{FL}$  the branch-and-cut method using this formulation. Some preliminary results that corroborate the previous observation are presented in table 6. We allow a maximum of 600 seconds CPU time for  $BC_{FL}$ .

| $N$     | $T$ | Method    | $UB$       | $LB$       | $NB_{Nodes}$ | $MIR_{Cuts}$ | $F_{Cuts}$ | $GAP$ |
|---------|-----|-----------|------------|------------|--------------|--------------|------------|-------|
| Class 1 |     |           |            |            |              |              |            |       |
| 6       | 15  | $BC_{FL}$ | 4 020 032  | 4 001 075  | 46 464       | 587          | 52         | 0,47% |
| 6       | 30  | $BC_{FL}$ | 4 378 291  | 4 279 466  | 5 623        | 1 068        | 70         | 2,26% |
| 12      | 15  | $BC_{FL}$ | 7 621 017  | 7 601 358  | 8 284        | 1 159        | 39         | 0,26% |
| 12      | 30  | $BC_{FL}$ | 8 528 870  | 8 335 720  | 8 040        | 556          | 44         | 2,26% |
| 24      | 15  | $BC_{FL}$ | 14 018 025 | 13 990 899 | 23 546       | 258          | 93         | 0,19% |
| 24      | 30  | $BC_{FL}$ | 23 344 574 | 23 250 108 | 7 200        | 168          | 80         | 0,40% |
| Class 2 |     |           |            |            |              |              |            |       |
| 6       | 15  | $BC_{FL}$ | 4 018 532  | 3 998 656  | 41 355       | 566          | 63         | 0,49% |
| 6       | 30  | $BC_{FL}$ | 4 420 710  | 4 275 341  | 6 608        | 1 059        | 72         | 3,29% |
| 12      | 15  | $BC_{FL}$ | 7 625 360  | 7 599 363  | 10 362       | 1 127        | 69         | 0,34% |
| 12      | 30  | $BC_{FL}$ | 8 537 235  | 8 333 043  | 9 389        | 491          | 45         | 2,39% |
| 24      | 15  | $BC_{FL}$ | 14 021 790 | 13 990 517 | 24 885       | 183          | 118        | 0,22% |
| 24      | 30  | $BC_{FL}$ | 23 333 497 | 23 249 672 | 5 835        | 109          | 81         | 0,36% |
| Class 3 |     |           |            |            |              |              |            |       |
| 6       | 15  | $BC_{FL}$ | 5 248 183  | 5 219 329  | 17 920       | 531          | 63         | 0,55% |
| 6       | 30  | $BC_{FL}$ | 7 033 213  | 6 750 190  | 3 161        | 872          | 186        | 4,02% |
| 12      | 15  | $BC_{FL}$ | 12 069 670 | 11 898 545 | 2 005        | 891          | 79         | 1,42% |
| 12      | 30  | $BC_{FL}$ | 15 210 791 | 14 603 870 | 473          | 1 346        | 150        | 3,99% |
| 24      | 15  | $BC_{FL}$ | 25 915 803 | 25 601 287 | 1 567        | 687          | 192        | 1,22% |
| 24      | 30  | $BC_{FL}$ | 43 265 295 | 41 630 487 | 190          | 1 160        | 334        | 3,78% |
| Class 4 |     |           |            |            |              |              |            |       |
| 6       | 15  | $BC_{FL}$ | 5 247 783  | 5 216 921  | 16 996       | 189          | 64         | 0,59% |
| 6       | 30  | $BC_{FL}$ | 6 991 352  | 6 742 206  | 3 601        | 804          | 228        | 3,56% |
| 12      | 15  | $BC_{FL}$ | 12 113 948 | 11 840 693 | 2 000        | 853          | 69         | 2,26% |
| 12      | 30  | $BC_{FL}$ | 15 477 755 | 14 196 428 | 290          | 1 222        | 97         | 8,28% |
| 24      | 15  | $BC_{FL}$ | 25 660 092 | 25 266 656 | 1 829        | 535          | 125        | 1,53% |
| 24      | 30  | $BC_{FL}$ | 41 504 711 | 39 151 154 | 130          | 566          | 173        | 5,67% |

Table 6: Computational results using the facility location-based formulation.

From table 6, we can easily notice that using the facility location-based formulation improves the performance of the  $BC$  method. Clearly  $BC_{FL}$  outperforms  $BC$ .

We also can notice that upper bounds obtained by  $BC_{FL}$  are better than those obtained by  $BC$  and  $BC+$ . Lower bounds of  $BC$  and  $BC_{FL}$  are almost equivalent.

We can also notice that the number of flow cover cuts generated by  $BC_{FL}$  is lower than the ones generated by  $BC$  and  $BC+$ , but the number of MIR cuts is greater for  $BC_{FL}$  than  $BC$  and  $BC+$ . This observation can be explained by the fact that we loose the flow structure induced by the flow conservation constraints in  $BC_{FL}$  when we use the facility location-based formulation.

Moreover, the number of nodes explored by the  $BC$  method is much more higher than the one explored by  $BC_{FL}$ . In fact, facility location-based formulation needs more variables and more constraints than the aggregated model. This new formulation may slows down the LP resolution at each node. Computational results show that to solve the LP relaxation of the facility location-



based formulation, we need an average of twice more CPU time than the LP relaxation of the aggregated formulation.

According to table 6, we can say that  $BC_{FL}$  is a promising method that can help us improving the branch-and-cut algorithm to solve production planning problems. Namely, it will be really interesting to generalize valid inequalities presented in this paper to the facility location-based formulation of MCLSSP problem and use them in a branch-and-cut framework.

## 10 Conclusion

We proposed a mathematical formulation of a new capacitated lot-sizing problem with setup times and shortage costs. A polyhedral approach has yielded strong valid inequalities. Computational experiments suggests that the use of these inequalities significantly improves the algorithms used to solve this kind of problems. There are many enhancement means to follow up these results. Namely, we study the polyhedral structure of the convex hull of the proposed model which helps us to prove that the cover inequalities induce facets of the convex hull under certain conditions [2]. By following the same approach, it could be useful to prove that reverse cover inequalities are also facet defining under certain conditions. The valid inequalities presented in this document were generalized to take into account other practical constraints that occur frequently in industrial situations, notably minimal production level and minimum run constraints. These inequalities were also generalized when more than one resource is available. Some extensions could be done when we have to deal with setup constraints on groups of items. From a scheduling perspective, these valid inequalities can be generalized to include start-up costs. We can quote Van Hoesel *et al.* [37]. They generalized the  $(l, S)$  inequalities to a new class of valid inequalities  $(l, R, S)$  to deal with start-up costs for the uncapacitated lot-sizing problem. It should be interesting to pursue this work to generalize the valid inequalities presented in this paper. The extension of the valid inequalities for the facility location-based formulation is also a promising track to enhance the effectiveness of the approach. Finally, it would be also interesting to use this approach in conjunction with a heuristic as the time decomposition based heuristic presented in [1].

## References

- [1] N. Absi and S. Kedad-Sidhoum. Une heuristique pour un problème de lot-sizing à capacité finie sous contraintes. In *Manifestation de JEunes Chercheurs STIC'04*, Calais, France, 2004. Université du littoral.
- [2] N. Absi and S. Kedad-Sidhoum. Multi-item capacitated lot-sizing problem with setup times and shortage costs : Polyhedral results. Technical Report LIP6 2005/009, LIP6, 2005.
- [3] D. Aksen, K. Altinkemer, and S. Chand. The single-item lot-sizing problem with immediate lost sales. *European Journal of Operational Research*, 147:558–566, 2003.
- [4] I. Barany, T. Van Roy, and L.A. Wolsey. Strong formulations for multi-item capacitated lot-sizing. *Management Science*, 30 (10):1255–1261, 1984.
- [5] I. Barany, T. Van Roy, and L.A. Wolsey. Uncapacitated lot-sizing : the convex hull of solutions. *Mathematical Programming Study*, 22:32–43, 1984.
- [6] G. Belvaux and L.A. Wolsey. bc-prod: A specialized branch-and-cut system for lot-sizing problems. *Management Science*, 46:724–738, 2000.
- [7] G. Bitran and H.H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28:1174–1186, 1982.
- [8] N. Brahimi. *Planification de la production: modèles et algorithmes pour les problèmes de dimensionnement de lots*. PhD thesis, Université de Nantes, 2004.
- [9] N. Brahimi, S. Dauzère-Pérès, N.M. Najid, and A. Nordli. Single item lot sizing problems. *European Journal of Operational Research*, 168 (1):1–16, 2006.
- [10] W.H. Chen and J.M. Thizy. Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research*, 26:29–72, 1990.

- [11] M. Constantino. Lower bounds in lot-sizing models: a polyhedral study. *Mathematics of Operations Research*, 23:101–118, 1998.
- [12] M. Diaby, H.C. Bahl, M.H. Karwan, and S. Zionts. A lagrangean relaxation approach for very large scale capacitated lot-sizing. *Management Science*, 38 (9):1329–1340, 1992.
- [13] P.S. Dixon, M.D. Elder, G.K. Rand, and E.A. Silver. A heuristic algorithm for determining lot sizes of an item subject to regular and overtime production capacities. *Journal of Operations Management*, 3:121–130, 1983.
- [14] J.R. Evans. An efficient implementation of the wagner-whitin algorithm for dynamic lot-sizing. *Journal of Operations Management*, 5 (2):229–235, 1985.
- [15] M. Florian, J.K. Lenstra, and A.H.G. Rinnoy Kan. Deterministic production planning : algorithms and complexity. *Management Science*, 26:669–679, 1980.
- [16] R. E. Gomory. An algorithm for the mixed integer problem. Technical report, The Rand Corporation, 1960.
- [17] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4:109–129, 2000.
- [18] J. Krarup and O. Bilde. Plant location, set covering and economic lot sizes: An  $O(mn)$ -algorithm for structured problems, in *optimierung bei graphentheoretischen und ganzzahligen probleme*. L. COLLATZ et al. (eds), Birkhauser Verlag, Basel, pages 155–180, 1977.
- [19] J.M. Leung, T.L. Magnanti, and R. Vachani. Facets and algorithms for capacitated lot-sizing. *Mathematical Programming*, 45:331–359, 1989.
- [20] CPLEX Callable Library. Ilog sa. <http://www.ilog.com>, 2005.
- [21] M. Loparic, Y. Pochet, and L.A. Wolsey. The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming*, 89:487–504, 2001.
- [22] LOTSIZELIB. <http://www.core.ucl.ac.be:16080/wolsey/lotsizel.htm>, 1999.
- [23] A.S. Manne. Programming of economic lot-sizes. *Management Science*, 4:115–135, 1958.
- [24] H. Marchand and L.A. Wolsey. The 0-1 knapsack problem with a single continuous variable. *Mathematical Programming*, 85:15–33, 1999.
- [25] A. J. Miller. *Polyhedral Approaches to Capacitated Lot-Sizing Problems*. PhD thesis, Georgia Institute of Technology, 1999.
- [26] A. J. Miller, G.L. Nemhauser, and M.W.P. Savelsbergh. On the polyhedral structure of a multi-item production planning model with setup times. *Mathematical Programming*, 94:375–405, 2003.
- [27] G. L. Nemhauser and L. A. Wolsey. A recursive procedure for generating all cuts for 0-1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990.
- [28] L. Özdamar and S.I. Birbil. Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research*, 110:525–547, 1998.
- [29] L. Özdamar and M.A. Bozyel. The capacitated lot sizing problem with overtime decisions and setup times. *IIE Transactions*, 32:1043–1057, 2000.
- [30] M.W. Padberg, T.J. Van Roy, and L.A. Wolsey. Valid linear inequalities for fixed charge problems. *Operations Research*, 33:842–861, 1985.
- [31] Y. Pochet and L.A. Wolsey. Lot-size models with backlogging: strong reformulations and cutting planes. *Mathematical Programming*, 40:317–335, 1988.
- [32] Y. Pochet and L.A. Wolsey. Solving multi-item lot-sizing problems using strong cutting planes. *Management Science*, 37:53–67, 1991.
- [33] Y. Pochet and L.A. Wolsey. Lot-sizing with constant batches: formulation and valid inequalities. *Mathematics of Operations Research*, 18:767–785, 1993.
- [34] T.J. Van Roy and L.A. Wolsey. Valid inequalities for mixed 0-1 programs. *Discrete Applied Mathematics*, 14:199–213, 1986.

- [35] R.A. Sandbothe and G.L. Thompson. A forward algorithm for the capacitated lot size model with stockouts. *Operations Research*, 38 (3):474–486, 1990.
- [36] W. Trigeiro, L.J. Thomas, and J.O. McLain. Capacitated lot-sizing with setup times. *Management Science*, 35:353–366, 1989.
- [37] C.P.M. van Hoesel, A.P.M. Wagelmans, and L.A. Wolsey. Polyhedral characterization of the economic lot-sizing problem with start-up costs. *Journal of Discrete Mathematics*, 7:141–151, 1994.
- [38] H.M. Wagner and T.M. Whitin. A dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.
- [39] L.A. Wolsey. Valid inequalities and superadditivity for 0/1 integer programs. *Mathematics of Operations Research*, 2:66–77, 1977.
- [40] L.A. Wolsey. *Integer Programming*. New York : Wiley, 1998.
- [41] W.I. Zangwill. A deterministic multi-period production scheduling model with backlogging. *Management Science*, 13:105–119, 1966.